

888888888888 0000000000 0000000000 TTTTTTTTTTTTTTTTT SSSSSSSSSSSSS
888888888888 0000000000 0000000000 TTTTTTTTTTTTTTTTT SSSSSSSSSSSSS
888888888888 0000000000 0000000000 TTTTTTTTTTTTTTTTT SSSSSSSSSSSSS
888 888 000 000 000 000 000 TTT TSS
888 888 000 000 000 000 000 TTT TSS
888 888 000 000 000 000 000 TTT TSS
888 888 000 000 000 000 000 TTT TSS
888 888 000 000 000 000 000 TTT TSS
888 888 000 000 000 000 000 TTT TSS
888 888 000 000 000 000 000 TTT TSS
888888888888 000 000 000 000 000 TTT SSSSSSSSS
888888888888 000 000 000 000 000 TTT SSSSSSSSS
888888888888 000 000 000 000 000 TTT SSSSSSSSS
888 888 000 000 000 000 000 TTT TSS
888 888 000 000 000 000 000 TTT TSS
888 888 000 000 000 000 000 TTT TSS
888 888 000 000 000 000 000 TTT TSS
888 888 000 000 000 000 000 TTT TSS
888 888 000 000 000 000 000 TTT TSS
888888888888 0000000000 0000000000 TTT SSSSSSSSSSSSS
888888888888 0000000000 0000000000 TTT SSSSSSSSSSSSS
888888888888 0000000000 0000000000 TTT SSSSSSSSSSSSS

BBBBBBBBBB	TTTTTTTTTT	MM	MM	EEEEEEEEE	MM	MM	888888	SSSSSSSS	SSSSSSSS
BBBBBBBBBB	TTTTTTTTTT	MM	MM	EEEEEEEEE	MM	MM	888888	SSSSSSSS	SSSSSSSS
BB	BB	TT	MMMM	MMMM	EE	MMMM	MMMM	88	88
BB	BB	TT	MMMM	MMMM	EE	MMMM	MMMM	88	88
BB	BB	TT	MM	MM	EE	MM	MM	88	88
BB	BB	TT	MM	MM	EE	MM	MM	88	88
BB	BB	TT	MM	MM	EEEEE	MM	MM	888888	SSSSSS
BB	BB	TT	MM	MM	EEEEE	MM	MM	888888	SSSSSS
BB	BB	TT	MM	MM	EE	MM	MM	88	SS
BB	BB	TT	MM	MM	EE	MM	MM	88	SS
BB	BB	TT	MM	MM	EE	MM	MM	88	SS
BB	BB	TT	MM	MM	EEEEE	MM	MM	888888	SSSSSS
BB	BB	TT	MM	MM	EEEEE	MM	MM	888888	SSSSSS
BB	BB	TT	MM	MM	EE	MM	MM	88	SS
BB	BB	TT	MM	MM	EE	MM	MM	88	SS
BB	BB	TT	MM	MM	EEEEE	MM	MM	888888	SSSSSS
BB	BB	TT	MM	MM	EEEEE	MM	MM	888888	SSSSSS

LL	IIIIII	SSSSSSSS
LL	IIIIII	SSSSSSSS
LL	II	SS
LLLLLLLL	IIIIII	SSSSSSSS
LLLLLLLL	IIIIII	SSSSSSSS

(2) 119
(3) 148

Declarations
CHECKMEM_8SS, Identify 11/8SS memory

0000 1 .TITLE BTMEM8SS - Configure and Test 11/8SS Memory
0000 2 .IDENT 'V04-000'
0000 3 .*****
0000 4 .
0000 5 .
0000 6 .
0000 7 .* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 .* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 .* ALL RIGHTS RESERVED.
0000 10 .*
0000 11 .* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 .* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 .* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 .* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 .* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 .* TRANSFERRED.
0000 17 .*
0000 18 .* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 .* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 .* CORPORATION.
0000 21 .*
0000 22 .* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 .* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 .*
0000 25 .*
0000 26 .*****
0000 27 .
0000 28 .++
0000 29 .
0000 30 .
0000 31 .
0000 32 .
0000 33 .
0000 34 .
0000 35 .
0000 36 .
0000 37 .
0000 38 .
0000 39 .
0000 40 .
0000 41 .
0000 42 .
0000 43 .
0000 44 .
0000 45 .
0000 46 .
0000 47 .
0000 48 .
0000 49 .
0000 50 .
0000 51 .
0000 52 .
0000 53 .
0000 54 .
0000 55 .
0000 56 .
0000 57 .
0000 58 .
0000 59 .
0000 60 .
0000 61 .
0000 62 .
0000 63 .
0000 64 .
0000 65 .
0000 66 .
0000 67 .
0000 68 .
0000 69 .
0000 70 .
0000 71 .
0000 72 .
0000 73 .
0000 74 .
0000 75 .
0000 76 .
0000 77 .
0000 78 .
0000 79 .
0000 80 .
0000 81 .
0000 82 .
0000 83 .
0000 84 .
0000 85 .
0000 86 .
0000 87 .
0000 88 .
0000 89 .
0000 90 .
0000 91 .
0000 92 .
0000 93 .
0000 94 .
0000 95 .
0000 96 .
0000 97 .
0000 98 .
0000 99 .
0000 100 .
0000 101 .
0000 102 .
0000 103 .
0000 104 .
0000 105 .
0000 106 .
0000 107 .
0000 108 .
0000 109 .
0000 110 .
0000 111 .
0000 112 .
0000 113 .
0000 114 .
0000 115 .
0000 116 .
0000 117 .
0000 118 .
0000 119 .
0000 120 .
0000 121 .
0000 122 .
0000 123 .
0000 124 .
0000 125 .
0000 126 .
0000 127 .
0000 128 .
0000 129 .
0000 130 .
0000 131 .
0000 132 .
0000 133 .
0000 134 .
0000 135 .
0000 136 .
0000 137 .
0000 138 .
0000 139 .
0000 140 .
0000 141 .
0000 142 .
0000 143 .
0000 144 .
0000 145 .
0000 146 .
0000 147 .
0000 148 .
0000 149 .
0000 150 .
0000 151 .
0000 152 .
0000 153 .
0000 154 .
0000 155 .
0000 156 .
0000 157 .
0000 158 .
0000 159 .
0000 160 .
0000 161 .
0000 162 .
0000 163 .
0000 164 .
0000 165 .
0000 166 .
0000 167 .
0000 168 .
0000 169 .
0000 170 .
0000 171 .
0000 172 .
0000 173 .
0000 174 .
0000 175 .
0000 176 .
0000 177 .
0000 178 .
0000 179 .
0000 180 .
0000 181 .
0000 182 .
0000 183 .
0000 184 .
0000 185 .
0000 186 .
0000 187 .
0000 188 .
0000 189 .
0000 190 .
0000 191 .
0000 192 .
0000 193 .
0000 194 .
0000 195 .
0000 196 .
0000 197 .
0000 198 .
0000 199 .
0000 200 .
0000 201 .
0000 202 .
0000 203 .
0000 204 .
0000 205 .
0000 206 .
0000 207 .
0000 208 .
0000 209 .
0000 210 .
0000 211 .
0000 212 .
0000 213 .
0000 214 .
0000 215 .
0000 216 .
0000 217 .
0000 218 .
0000 219 .
0000 220 .
0000 221 .
0000 222 .
0000 223 .
0000 224 .
0000 225 .
0000 226 .
0000 227 .
0000 228 .
0000 229 .
0000 230 .
0000 231 .
0000 232 .
0000 233 .
0000 234 .
0000 235 .
0000 236 .
0000 237 .
0000 238 .
0000 239 .
0000 240 .
0000 241 .
0000 242 .
0000 243 .
0000 244 .
0000 245 .
0000 246 .
0000 247 .
0000 248 .
0000 249 .
0000 250 .
0000 251 .
0000 252 .
0000 253 .
0000 254 .
0000 255 .
0000 256 .
0000 257 .
0000 258 .
0000 259 .
0000 260 .
0000 261 .
0000 262 .
0000 263 .
0000 264 .
0000 265 .
0000 266 .
0000 267 .
0000 268 .
0000 269 .
0000 270 .
0000 271 .
0000 272 .
0000 273 .
0000 274 .
0000 275 .
0000 276 .
0000 277 .
0000 278 .
0000 279 .
0000 280 .
0000 281 .
0000 282 .
0000 283 .
0000 284 .
0000 285 .
0000 286 .
0000 287 .
0000 288 .
0000 289 .
0000 290 .
0000 291 .
0000 292 .
0000 293 .
0000 294 .
0000 295 .
0000 296 .
0000 297 .
0000 298 .
0000 299 .
0000 300 .
0000 301 .
0000 302 .
0000 303 .
0000 304 .
0000 305 .
0000 306 .
0000 307 .
0000 308 .
0000 309 .
0000 310 .
0000 311 .
0000 312 .
0000 313 .
0000 314 .
0000 315 .
0000 316 .
0000 317 .
0000 318 .
0000 319 .
0000 320 .
0000 321 .
0000 322 .
0000 323 .
0000 324 .
0000 325 .
0000 326 .
0000 327 .
0000 328 .
0000 329 .
0000 330 .
0000 331 .
0000 332 .
0000 333 .
0000 334 .
0000 335 .
0000 336 .
0000 337 .
0000 338 .
0000 339 .
0000 340 .
0000 341 .
0000 342 .
0000 343 .
0000 344 .
0000 345 .
0000 346 .
0000 347 .
0000 348 .
0000 349 .
0000 350 .
0000 351 .
0000 352 .
0000 353 .
0000 354 .
0000 355 .
0000 356 .
0000 357 .
0000 358 .
0000 359 .
0000 360 .
0000 361 .
0000 362 .
0000 363 .
0000 364 .
0000 365 .
0000 366 .
0000 367 .
0000 368 .
0000 369 .
0000 370 .
0000 371 .
0000 372 .
0000 373 .
0000 374 .
0000 375 .
0000 376 .
0000 377 .
0000 378 .
0000 379 .
0000 380 .
0000 381 .
0000 382 .
0000 383 .
0000 384 .
0000 385 .
0000 386 .
0000 387 .
0000 388 .
0000 389 .
0000 390 .
0000 391 .
0000 392 .
0000 393 .
0000 394 .
0000 395 .
0000 396 .
0000 397 .
0000 398 .
0000 399 .
0000 400 .
0000 401 .
0000 402 .
0000 403 .
0000 404 .
0000 405 .
0000 406 .
0000 407 .
0000 408 .
0000 409 .
0000 410 .
0000 411 .
0000 412 .
0000 413 .
0000 414 .
0000 415 .
0000 416 .
0000 417 .
0000 418 .
0000 419 .
0000 420 .
0000 421 .
0000 422 .
0000 423 .
0000 424 .
0000 425 .
0000 426 .
0000 427 .
0000 428 .
0000 429 .
0000 430 .
0000 431 .
0000 432 .
0000 433 .
0000 434 .
0000 435 .
0000 436 .
0000 437 .
0000 438 .
0000 439 .
0000 440 .
0000 441 .
0000 442 .
0000 443 .
0000 444 .
0000 445 .
0000 446 .
0000 447 .
0000 448 .
0000 449 .
0000 450 .
0000 451 .
0000 452 .
0000 453 .
0000 454 .
0000 455 .
0000 456 .
0000 457 .
0000 458 .
0000 459 .
0000 460 .
0000 461 .
0000 462 .
0000 463 .
0000 464 .
0000 465 .
0000 466 .
0000 467 .
0000 468 .
0000 469 .
0000 470 .
0000 471 .
0000 472 .
0000 473 .
0000 474 .
0000 475 .
0000 476 .
0000 477 .
0000 478 .
0000 479 .
0000 480 .
0000 481 .
0000 482 .
0000 483 .
0000 484 .
0000 485 .
0000 486 .
0000 487 .
0000 488 .
0000 489 .
0000 490 .
0000 491 .
0000 492 .
0000 493 .
0000 494 .
0000 495 .
0000 496 .
0000 497 .
0000 498 .
0000 499 .
0000 500 .
0000 501 .
0000 502 .
0000 503 .
0000 504 .
0000 505 .
0000 506 .
0000 507 .
0000 508 .
0000 509 .
0000 510 .
0000 511 .
0000 512 .
0000 513 .
0000 514 .
0000 515 .
0000 516 .
0000 517 .
0000 518 .
0000 519 .
0000 520 .
0000 521 .
0000 522 .
0000 523 .
0000 524 .
0000 525 .
0000 526 .
0000 527 .
0000 528 .
0000 529 .
0000 530 .
0000 531 .
0000 532 .
0000 533 .
0000 534 .
0000 535 .
0000 536 .
0000 537 .
0000 538 .
0000 539 .
0000 540 .
0000 541 .
0000 542 .
0000 543 .
0000 544 .
0000 545 .
0000 546 .
0000 547 .
0000 548 .
0000 549 .
0000 550 .
0000 551 .
0000 552 .
0000 553 .
0000 554 .
0000 555 .
0000 556 .
0000 557 .
0000 558 .
0000 559 .
0000 560 .
0000 561 .
0000 562 .
0000 563 .
0000 564 .
0000 565 .
0000 566 .
0000 567 .
0000 568 .
0000 569 .
0000 570 .
0000 571 .
0000 572 .
0000 573 .
0000 574 .
0000 575 .
0000 576 .
0000 577 .
0000 578 .
0000 579 .
0000 580 .
0000 581 .
0000 582 .
0000 583 .
0000 584 .
0000 585 .
0000 586 .
0000 587 .
0000 588 .
0000 589 .
0000 590 .
0000 591 .
0000 592 .
0000 593 .
0000 594 .
0000 595 .
0000 596 .
0000 597 .
0000 598 .
0000 599 .
0000 600 .
0000 601 .
0000 602 .
0000 603 .
0000 604 .
0000 605 .
0000 606 .
0000 607 .
0000 608 .
0000 609 .
0000 610 .
0000 611 .
0000 612 .
0000 613 .
0000

0000 58 : IMPLICIT INPUTS:

0000 59

0000 60 The positions on the 11/8SS system bus, the BI, are called nodes, and

0000 61 are identified by node numbers 0-15.

0000 62

0000 63 BI I/O space begins at ^x20000000 (512MB), and each node occupies

0000 64 an 8KB node space of registers.

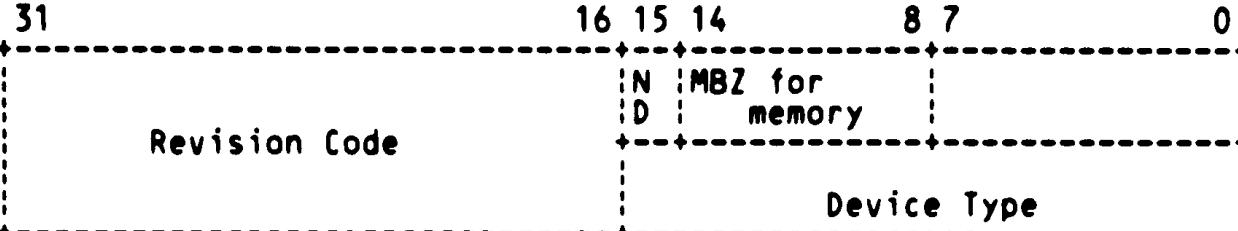
0000 65

0000 66 BI nodes are self identifying in that the first register (relative 0)

0000 67 in the node space is constrained to be the DEVTYPE register. The

0000 68 format of this register is as follows:

0000 69

0000 70 

0000 71

0000 72

0000 73

0000 74 Revision Code

0000 75

0000 76

0000 77

0000 78

0000 79 This register is to be interpreted as follows:

0000 80

0000 81 The Device Type field is broken up into three sub-fields.

0000 82 The main sub-field of interest here is the one defined

0000 83 by bits $\langle 14:08 \rangle$. If this sub field is zero then this

0000 84 node is a BI memory node. All other types of node,

0000 85 i.e. processors, adapters, etc., are constrained to

0000 86 have non-zero values here.

0000 87

0000 88 The ND field, bit $\langle 15 \rangle$, if non-zero, indicates that the

0000 89 node is a non-DEC device.

0000 90

0000 91

0000 92 : OUTPUTS:

0000 93

0000 94 R7, R8, R11, and SP are preserved

0000 95 All others (including AP and FP) are altered

0000 96

0000 97 : IMPLICIT OUTPUTS:

0000 98

0000 99 The PFN bitmap is modified to describe all of physical memory.

0000 100

0000 101 RPB\$L_PFN_CNT stores the number of pages of physical memory.

0000 102

0000 103

0000 104

0000 105

0000 106 AUTHOR:

0000 107 Robert L. Rappaport, creation date 10-Oct-1983

0000 108

0000 109 : REVISION HISTORY:

0000 110 V03-002 RLR0001 Robert L. Rappaport 6-Apr-1984

0000 111 Several modifications. 1) Test for BROKE bit, and do NOT

0000 112 test memory if set; 2) Clear error bits in CSR1 and CSR2;

0000 113 3) move replacement of Machine Check handler from loop;

0000 114 4) PUSHL original Machine Check handler addr and POPL it

0000 115 : after loop.
0000 116 : -
0000 117 : -

0000 119 .SBTTL Declarations
0000 120 .DEFAULT DISPLACEMENT, WORD
0000 121
0000 122
0000 123 :
0000 124 : Macros to describe VMS data structures
0000 125 :
0000 126 :
0000 127 \$BIICDEF : BIIC registers
0000 128 \$BIMEMDEF : BI Memory registers
0000 129 \$DMPDEF : System dump file header definitions
0000 130 \$IO8SSDEF : 11/8SS definitions
0000 131 \$NDTDEF : Nexus device types
0000 132 \$PRDEF : Processor registers
0000 133 \$PR8SSDEF : CPU Specific Processor registers
0000 134 \$RPBDEF : Restart parameter block
0000 135 \$BUADEF : BI-UNIBUS adapter
0000 136
0000 137 :
0000 138 : Macros
0000 139 :
0000 140 :
0000 141 .MACRO ERROR,STR : Outputs an error string to the
0000 142 BSBW ERROUT : console terminal.
0000 143 .ASCIZ STR
0000 144 .ENDM ERROR
0000 145
00000000 146 .PSECT YBTMEM, LONG

0000 148 .SBTTL CHECKMEM_8SS, Identify 11/8SS memory
 0000 149
 0000 150 :++
 0000 151 : CHECKMEM_8SS, Locate and test memory for 11/8SS
 0000 152 :--
 0000 153 :
 0000 154 :
 0000 155 :
 0000 156 CHECKMEM_8SS:: ; Entry for 11/8SS.
 0000 157 :
 0000 158 :
 0000 159 : Start testing slot positions to find adapters. First save the stack
 0000 160 : position so it can be restored after a machine check.
 0000 161 :
 0000 162 :
 0000 163 INIT_SEARCH: ; Start searching for adapters
 0000 164 :
 0000 165 : Save original machine check handler address on stack.
 0000 166 :
 0000 167 SD 04 A7 DD 0000 167 PUSHL 4(R7) ; Save original handler address.
 5D SE 0003 168 MOVL SP,FP ; Save current top of stack.
 0006 169
 0006 170 :
 0006 171 : Set up the physical address of the 1st slot on the system bus and
 0006 172 : the address of the adapter type table.
 0006 173 :
 0006 174 :
 54 20000000 9F 9E 0006 175 MOVAB #IO8SSAL_I0BASE,R4 ; Get address of 1st slot.
 SC 10 9A 0000 176 MOVZBL #IO8SSAL_NNEX,AP ; Set up NEXUS loop counter.
 0010 177
 0010 178 :
 0010 179 : During this memory locate and test loop, the following registers are
 0010 180 : used:
 0010 181 :
 0010 182 : R0 - the contents of the slot's configuration register
 0010 183 : R4 - address of the configuration register at the current
 0010 184 : slot position
 0010 185 : R7 - address of the SCB
 0010 186 : R9 - bit setting in memory present map;
 0010 187 : starting page number in this controller
 0010 188 : R10 - address of the memory description list in RPB (pagcnt & pfn)
 0010 189 : R11 - address of the RPB
 0010 190 :
 0010 191 : Initialize R10 (RPB memory descriptor list pointer) for search loop
 0010 192 :
 30 AB 5A 00BC CB 9E 0010 193 MOVAB RPBSL MEMDSC(R11),R10 ; Set pointer to memory description list
 0000 0005800 8F CA 0015 194 BICL #<RPBSM_MPM ! RPBSM_USEMPM ! RPBSM_FINDMEM>, -
 001D 195 RPBSL_BOOTRS(R11) ; Clear all MA780-specific boot flags
 001D 196
 001D 197
 001D 198 TRY_NEXUS_8SS: ; Memory locate and test loop.
 001D 199
 001D 200 : Set up a machine check fault handler to gain control if the main loop
 001D 201 : here addresses a non-existent configuration register (an empty slot).
 001D 202 : Note we do this within the loop since we re-set to another machine
 001D 203 : check handler later on in the loop.
 001D 204

04 A7 6D'AF 9E 001D 205 MOVAB B^DO_NEXT_8SS+1,4(R7) ; Set up fault handler (+1 for
0022 206 ; handler execution on the
0022 207 ; interrupt stack).
0022 208
0022 209 ; Then read the slot's configuration register.
0022 210
50 64 D0 0022 211 MOVL (R4),R0 ; Read CR at current slot.
0025 212
0025 213 : Execution continues here if the configuration register is present.
0025 214 : If the adapter type is a memory controller, proceed to test memory.
0025 215 : Otherwise, move to the next BI slot.
0025 216
0025 217 :
0025 218
0025 219
50 50 08 EF 0025 220 EXT/V #BIICSV_MEMNODE,- ; Extract sub-field of BI devtype
0027 221 #BIICSS_MEMNODE,R0,R0 ; field that tells if we have memory.
40 12 002A 222 BNEQ DO_NEXT_8SS ; No, advance to next slot.
002C 223
002C 224 : Here check BROKE bit to see if memory passed self test.
002C 225
50 0100 C4 D0 002C 226 MOVL BIMEMSL_CSR1(R4),R0 ; Read CSR containing BROKE bit.
0031 227 BBS #BIMEM\$V_BROKE,- ; If BROKE, ignore this node.
37 50 0033 228
0035 229
0035 230 : Memory controller found:
0035 231
50 20 A4 D1 0035 232 MOVL BIICSL_SAR(R4),R0 ; R0 = Starting Address for this memory.
51 24 A4 D1 0039 233 MOVL BIICSL_EAR(R4),R1 ; R1 = Ending Address for this memory.
50 17 9C 003D 234 ROTL #<32-95,R0,R9 ; Equivalent of ASHL #9,R0,R9. Result
0041 235 is starting PFN to R9.
53 51 50 C3 0041 236 SUBL3 R0,R1,R3 ; R3 = length of memory segment in bytes.
53 53 17 9C 0045 237 ROTL #<32-9>,R3,R3 ; Equivalent of ASHL #9,R3,R3. Result
0049 238 is # of pages to R3.
0049 239
8A 53 D0 0049 240 MOVL R3,(R10)+ ; Save # of pages in this memory
004C 241 ASSUME DMP\$V-TR EQ 24
004C 242 ASSUME DMP\$S-TR EQ 8
FF AA 10 90 004C 243 MOVB #108SS\$AL_NNEX,-1(R10) ; Compute the TR number for this
FF AA SC 82 0050 244 SUBB AP,-1(R10) ; memory and store in descriptor
8A 59 D0 0054 245 MOVL R9,(R10)+ ; Save starting PFN for this memory
0057 246
0057 247 : Before starting memory test, establish a page skipping handler for
0057 248 : machine checks, and turn off the cache so that writes followed by
0057 249 : reads to memory don't write to memory and then read from the cache.
0057 250 : Also, enable CRD error reporting if requested by the RPB BOOTRS flag.
0057 251
0057 252 :
0057 253
04 A7 00B9'CF 9E 0057 254 MOVAB PAGE_MCHECK_8SS+1,4(R7) ; Set page skipping handler (+1
005D 255 ; for interrupt stack).
01 DA 005D 256 MTPR #PR8SS\$M_CADR_D,- ; Turn off memory cache.
25 005F 257
52 0088'CF 9E 0060 258 MOVAB TEST QUAD_8SS,R2 ; Page test routine address
FF98' 30 0065 259 BSBW BOOSTTEST_MEM ; Test the specified range of PFN's
02 11 0068 260 BRB DO_NEXT_8SS ; Do the next controller if any
006A 261

006A 262 .ALIGN LONG ; Longword-aligned handler.
 006C 263
 006C 264
 006C 265 : Fault handler for non-existent configuration register, or unreadable
 006C 266 : registers, or a non-memory controller slot device. Restore stack
 006C 267 : pointer, clear all errors, and try for another slot if any remain.
 006C 268
 006C 269
 006C 270 DO_NEXT_8SS: ; Skip to next slot.
 26 5E 5D 00 006C 271 MOVL FP, SP ; Restore stack pointer.
 54 FFFFFFFF 8F DA 006F 272 MTPR #1, #PRS_MCESR ; Clear any faults.
 2000 C4 9E 0076 273 MOVAB IO8SS\$AL_PERNEX(R4), R4 ; Move to next slot.
 9F 5C F5 007B 274 SOBGTR AP, TRY_NEXUS_8SS ; If still a slot, loop.
 007E 275
 007E 276
 007E 277 : Reestablish the normal machine check fault handler.
 007E 278
 007E 279
 007E 280 ASSUME PR8SS\$V_CADR_D EQ 0
 25 00 DA 007E 281 MTPR #0, #PR8SS_CADR ; Re-enable cache.
 04 A7 8ED0 0081 282 POPL 4(R7) ; Restore original handler address.
 BA D4 0085 283 CLRL (R10)+ ; Indicate end of RPB memory descr list
 05 0087 284 RSB ; Return to main routine.
 0088 285
 0088 286 ++
 0088 287
 0088 288 : Functional Description:
 0088 289 Test a page of 8SS memory.
 0088 290
 0088 291
 0088 292 : Calling Sequence:
 0088 293 JSB TEST_QUAD_8SS
 0088 294
 0088 295 : Inputs:
 0088 296
 0088 297 R0 = starting address to test
 0088 298 R1 = Quad word iteration count (64)
 0088 299 R4 => Node Registers
 0088 300 R11 = Address of RPB
 0088 301
 0088 302
 0088 303 : Outputs:
 0088 304
 0088 305 Returns via RSB if the entire page is OK
 0088 306 Error exit via Machine Check code to BO0\$PAGE_MCHECK
 0088 307
 0088 308 :--
 0088 309
 0088 310 TEST_QUAD_8SS: ; Test 1 quadword at a time.
 0088 311 CCRQ (R0) ; Clear a quadword.
 80 60 7C 008A 312 CMPL (R0)+, (R0)+ ; Read both longwords, and
 008D 313
 008D 314
 008D 315
 008D 316 : If no gross errors occur in the clear to the quadword or in the
 008D 317 : subsequent read instruction, then execution continues below. Otherwise
 008D 318 : execution goes to the fault handler.

```

008D 319 :
008D 320
F8 51 F5 008D 321 SOBGTR R1,TEST_QUAD_8SS
0100 C4 D0 0090 322 MOVL BIMEMSL(CSR1(R4)),R1 ; Continue clearing unless done.
0100 C4 51 D0 0095 323 MOVL R1,BIMEMSL(CSR1(R4)) ; Read memory CSR1.
51 0104 C4 D0 009A 324 MOVL BIMEMSL(CSR2(R4)),R1 ; Re-write (write 1 to clear) CSR1.
0104 C4 51 D0 009F 325 MOVL R1,BIMEMSL(CSR2(R4)) ; Read memory CSR2 and retain value
10 E1 00A4 326 BBC #RPB$V CRDTEST- in R1 for later testing CRD bit.
01 30 AB 00A6 327 RPB$V CRDTEST- ; Re-write (write 1 to clear) CSR2.
05 00A9 328 RPB$V CRDTEST- ; Branch if CRD test is requested.
00AA 329 5$: RSB
00AA 330
00AA 331 : Check if a CRD error occurred on this page.
00AA 332
00AA 333 10$: RSB
00AA 334 BITL #BIMEMSM_CRDLOGR,R1 ; Test if CRD errors encountered.
F6 13 00B1 335 BEQL 5$ ; Branch if no CRD error occurred.
1E 11 00B3 336 BRB ERR_EXIT_8SS ; Else take error path.
00B5 337
00B5 338 .ALIGN LONG ; All handlers longword-aligned.
00B8 339
00B8 340
00B8 341 : Handler that gains control when a page has gross memory errors. Just
00B8 342 : clear the error, recover the stack top, and advance to the next page.
00B8 343
00B8 344
00B8 345 PAGE_MCHECK_8SS: ; Handle machine check.
51 0100 C4 D0 00B8 346 MOVE BIMEMSL(CSR1(R4)),R1 ; Read memory CSR1.
0100 C4 51 D0 00BD 347 MOVL R1,BIMEMSL(CSR1(R4)) ; Re-write (write 1 to clear) CSR1.
51 0104 C4 D0 00C2 348 MOVL BIMEMSL(CSR2(R4)),R1 ; Read memory CSR2.
0104 C4 51 D0 00C7 349 MOVL R1,BIMEMSL(CSR2(R4)) ; Re-write (write 1 to clear) CSR2.
26 FFFFFFFF 8F DA 00CC 350 MTPR #-1,#PRS_MCESR ; Clear error indicator.
00D3 351 ERR_EXIT_8SS: BRW BOOSPAGE_MCHECK ; Exit to common bad page code
FF2A' 31 00D3 352 .END
00D6 353

```

BTMEM8SS
Symbol table

- Configure and Test 11/8SS Memory

M 11

15-SEP-1984 23:44:25 VAX/VMS Macro V04-00
4-SEP-1984 23:03:23 [BOOTS.SRC]BTMEM8SS.MAR;1

Page 9 (3)

BIICSL_EAR	= 00000024
BIICSL_SAR	= 00000020
BIICSS_MEMNODE	= 00000007
BIICSV_MEMNODE	= 00000008
BIMEMSL_CSR1	= 00000100
BIMEMSL_CSR2	= 00000104
BIMEMSM_CRDLOGR	= 20000000
BIMEMSV_BROKE	= 0000000C
BOOSPAGE_MCHECK	***** X 02
BOOSTEST_MEM	***** X 02
CHECKMEM_8SS	00000000 RG 02
DMPSS_TR	= 00000008
DMPSV_TR	= 00000018
DO_NEXT_8SS	0000006C R 02
ERR_EXIT_8SS	000000D3 R 02
INIT_SEARCH	00000000 R 02
IO8SS\$AL_IOPAGE	= 20000000
IO8SS\$AL_NNEX	= 00000010
IO8SS\$AL_PERNEX	= 00002000
PAGE_MCHECK_8SS	00000088 R 02
PRS_ACESR	= 00000026
PR8SS\$M_CADR_D	= 00000001
PR8SS\$V_CADR_D	= 00000000
PR8SS\$_CADR	= 00000025
RPBSL_BOOTRS	= 00000030
RPBSL_MEMDSC	= 0000008C
RPBSM_FINDMEM	= 00004000
RPBSM_MPM	= 00000800
RPBSM_USEMPM	= 00001000
RPBSV_CRDTEST	= 00000010
TEST_QUAD_8SS	00000088 R 02
TRY_NEXUS_8SS	00000010 R 02

+-----+
! Psect synopsis !
+-----+

PSECT name

	Allocation	PSECT No.	Attributes													
ABS .	00000000	(0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE		
\$ABSS	00000000	(0.)	01 (1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE		
YBTMEM	00000006	(214.)	02 (2.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	LONG		

+-----+
! Performance indicators !
+-----+

Phase

	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.07	00:00:01.81
Command processing	118	00:00:00.59	00:00:06.13
Pass 1	234	00:00:06.50	00:00:15.60
Symbol table sort	0	00:00:00.87	00:00:01.69
Pass 2	75	00:00:01.37	00:00:02.74
Symbol table output	5	00:00:00.05	00:00:00.05
Psect synopsis output	1	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00

Assembler run totals 464 00:00:09.48 00:00:28.06

The working set limit was 1350 pages.

35196 bytes (69 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 631 non-local and 2 local symbols.
353 source lines were read in Pass 1, producing 13 object records in Pass 2.
18 pages of virtual memory were used to define 17 macros.

! Macro library statistics !

Macro library name

Macros defined

-\$255\$DUA28:[BOOTS.OBJ]BOOTS.MLB;1
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2
TOTALS (all libraries)

0
6
7
13

719 GETS were required to define 13 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:BTMEM8SS/OBJ=OBJ\$:BTMEM8SS MSRC\$:BTMEM8SS/UPDATE=(ENH\$:BTMEM8SS)+EXECMLS/LIB+LIB\$:BOOTS.MLB/LIB

0037 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CONFIG
LIS

BTMEM855
LIS

BTMEM790
LIS

CONFIGURE
LIS

BOOTDEF
LIS

BOOTIO
LIS

BOOTDRIVR
LIS

BOOTBLOCK
LIS

BTMEM200
LIS

BTMEM250
LIS

BTMEM280
LIS

CONFIGMN
LIS